



TAMPEREEN
AMMATTIKORKEAKOULU

Netvisor-Odoo-integraatio

Jaakko Komulainen

Opinnäytetyö
Marraskuu 2016
Tietojenkäsittelyn koulutusohjelma



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma

KOMULAINEN, JAAKKO:
Netvisor-Odoo-integraatio

Opinnäytetyö 26 sivua
Marraskuu 2016

Ohjelmistokehityksessä integraatio voidaan määritellä prosessiksi, jonka tuloksena kaksi tai useampi sovellus saadaan jakamaan tietoa keskenään. Ohjelmistointegraatioita tarvitaan usein tilanteissa, joissa yritys on siirtymässä uuteen järjestelmään. Siirtymävaiheessa on yleistä, että täytyy ylläpitää samaa tietoa kahdessa eri järjestelmässä. Ohjelmistointegraatiot voidaan jakaa karkeasti kolmeen eri kategoriaan: data-, viesti- tai prosessitason integraatioihin. Tässä työssä tehdään viestitason integraatio Netvisor-taloushallintojärjestelmän ja Odoo-toiminnanohjausjärjestelmän välillä. Työn minimivaatimuksena oli myyntilaskujen lähettäminen Odoosta Netvisoriin. Opinnäytetyön tuotteena syntyi toiminnanohjausjärjestelmään asennettava moduuli, joka täyttää työlle asetetun tavoitteen.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Information Systems

KOMULAINEN, JAAKKO:
Netvisor-Odoo-Integration

Bachelor's thesis 26 pages
November 2016

System integration can be defined as a process of linking two or more systems together which enables sharing data between the linked systems. System integration is often needed when companies are transitioning from one system into a new one. Its common that you need to maintain the same data in two systems in the transitional phase. System integration can be separated in three different levels of integration: data, message and process level.

This thesis is about message level integration between the Netvisor financial administration system and the Odoo ERP system. The minium requirement for this thesis was to create a functionality for sending sales invoices from Odoo to Netvisor.

As a result of this study, an ERP module was created. It met all requirements that was set for it.

SISÄLLYS

| | |
|---|-----------|
| <u>1 JOHDANTO.....</u> | <u>6</u> |
| <u>2 OHJELMISTOINTEGRAATIOT.....</u> | <u>7</u> |
| <u>2.1. Ohjelmistointegraation tasot.....</u> | <u>7</u> |
| <u>2.2. Integraatiototeutuksen suunnittelu ja valmistelu.....</u> | <u>8</u> |
| <u>2.3. Avoimet integraatorajapinnat.....</u> | <u>8</u> |
| <u>3 ODOO-TOIMINNANOHJAUSJÄRJESTELMÄ.....</u> | <u>10</u> |
| <u>3.1. Järjestelmän kuvaus.....</u> | <u>10</u> |
| <u>3.2. Odoon arkkitehtuuri.....</u> | <u>11</u> |
| <u>3.2.1 Tietokantataso.....</u> | <u>11</u> |
| <u>3.2.2 Sovellustaso.....</u> | <u>11</u> |
| <u>3.2.3 Näkymätaso.....</u> | <u>11</u> |
| <u>3.3. ORM-moottori.....</u> | <u>12</u> |
| <u>4 NETVISOR-TALOUSHALLINTOPALVELU.....</u> | <u>15</u> |
| <u>4.1. Palvelut.....</u> | <u>15</u> |
| <u>4.2. Tiedonsiirto.....</u> | <u>16</u> |
| <u>4.3. Verkkolaskut.....</u> | <u>16</u> |
| <u>4.4. Tietoturva.....</u> | <u>16</u> |
| <u>4.5. Ohjelmistorajapinta.....</u> | <u>17</u> |
| <u>4.5.1 Tietojen tuominen järjestelmään.....</u> | <u>17</u> |
| <u>4.5.2 Tietojen noutaminen järjestelmästä.....</u> | <u>17</u> |
| <u>4.5.3 Yhteydenottotapa.....</u> | <u>18</u> |
| <u>4.5.4 Tietoturva.....</u> | <u>18</u> |
| <u>4.5.5 Rajapintakirjasto.....</u> | <u>18</u> |
| <u>5 INTEGRAATION SUUNNITTELU JA TOTEUTUS.....</u> | <u>19</u> |
| <u>5.1. Integraation suunnittelu.....</u> | <u>19</u> |
| <u>5.2. Toteutus.....</u> | <u>19</u> |
| <u>5.2.1 Asetukset.....</u> | <u>19</u> |
| <u>5.2.2 Laskun vienti Odoosta Netvisoriin.....</u> | <u>19</u> |
| <u>5.2.3 XML-laskuaineiston luonti sekä tarkistus.....</u> | <u>22</u> |
| <u>5.2.4 Aineiston lähettäminen Netvisor-rajapintaan.....</u> | <u>23</u> |
| <u>6 POHDINTA.....</u> | <u>25</u> |
| <u>LÄHTEET.....</u> | <u>26</u> |

LYHENTEET JA TERMIT

| | |
|---------------|--|
| Netvisor | Taloushallinto- ja laskutuspalvelu |
| Odoo, OpenERP | Toiminnanohjausjärjestelmä |
| ERP | Enterprise Resource Planning, toiminnanohjausjärjestelmä |
| CRM | Customer Relationship Management, asiakkuudenhallinta |
| Python | Ohjelmointikieli |
| SOAP | Simple Object Access Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| REST | Representational State Transfer |
| ORM | Object-Relational Mapping |
| API | Application Programmin Interface, ohjelmointirajapinta |
| SQL | Structured Query Language |
| XML | Extensible Markup Language, merkinäkieli |
| Finvoice | XML-muotoinen verkkolasku |
| MAC-koodi | Message Authentication Code, viestin turvatarkiste |

1 JOHDANTO

Opinnäytetyön tarkoituksena on luoda toiminnallisuus tiedon lähettämiseen Odoo toiminnanohjausjärjestelmän sekä Netvisor-taloushallintojärjestelmän välille. Työn toimeksiantajana toimii tamperelainen Vizucom Oy, joka tuottaa avoimen lähdekoodin toiminnanohjaus- ja auditointijärjestelmiä. Työn tavoitteena on selvittää kuinka Netvisorin tarjoamaa rajapintaa voidaan käyttää laskuaineiston tuomiseen toisesta järjestelmästä ja selvittää integraation ongelmakohtia. Työn lopputuloksena on tuottaa toimeksiantajalle Odoo-moduuli, jota voidaan jatkossa kehittää tai käyttää apuna muissa ohjelmistointegraatioissa. Raportin ensimmäisessä kappaleessa käydään läpi yleisesti ohjelmistointegraation pääperiaatteita. Seuraavissa kappaleissa annetaan kuvaukset järjestelmistä sekä niiden olleellisista osista integraatiossa. Järjestelmäkuvausten jälkeen käydään läpi teknistä toteutusta ja selvitetään mitä tulee ottaa huomioon laskuaineiston siirtämisessä Odoosta Netvisoriin.

2 OHJELMISTOINTEGRAATIOT

Ohjelmistokehityksessä integraatio voidaan määritellä prosessiksi, jonka tuloksena kaksi tai useampi sovellusta saadaan jakamaan tietoa keskenään (Nykänen, Pirkko 2015). Ohjelmistointegraatioille on useita syitä. Saman tiedon syöttäminen kahteen eri järjestelmään ei ole tehokasta. Myös inhimillisten virheiden mahdollisuudet vähenevät, kun esimerkiksi verkkokaupan tilaustietoja ei kirjoiteta verkkokauppasovelluksen lisäksi erilliseen laskutusjärjestelmään (Malinen, Teemu 2013).

Järjestelmien käyttöönottojen yhteydessä tarvitaan usein integraatioita vanhan ja uuden järjestelmän välille. Esimerkiksi toiminnanohjausjärjestelmän käyttöönotto tarkoittaa usein, että tietoa tuodaan useista eri lähteistä toiminnanohjausjärjestelmään. Siirtyminen järjestelmästä toiseen ei useinkaan tapahdu niin sanotusti kertarysäyksellä, vaan toiminnanohjausjärjestelmän käyttöönotto on vaiheittainen. Esimerkiksi kirjanpidon raportointi edellyttää tiedon siirtämistä toiminnanohjausjärjestelmästä vanhaan järjestelmään sekä toisin päin.

2.1. Ohjelmistointegraation tasot

Integraatio voidaan jakaa seuraaviin tasoihin (Nykänen, Pirkko 2015):

- Data-taso
 - Kohteena tietokannat
 - Asetetaan järjestelmät käyttämään jaettuja tietokantoja
 - Tietokantojen väliset tiedonsiirrot
- Viestitaso
 - Tiedonsiirto viestien välityksellä
 - Sanomarakenteen määrittely sekä standardointi
 - Ohjelmointirajapinnat
- Prosessitaso
 - Yrityksen liiketoimintaprosessit
 - Prosessien määrittely

Tässä opinnäytetyössä toteutetaan viestitason integraatiota. Netvisorin tarjoaman ohjelmointirajapinnan, eli API:n kautta voidaan lähettää sekä noutaa tietoa järjestelmien välillä (Netvisor: rajapintakuvaus 2015). Tarkemmin Netvisorin API:sta luvussa 4.5.

2.2. Integraatiototeutuksen suunnittelu ja valmistelu

Ohjelmistointegraatioita suunnitellessa on tärkeää ottaa huomioon integraatiokehityksessä mukana olevien järjestelmien rajoitukset sekä spesifikaatiot (Nykänen, Pirkko 2015). Siirrettäessä tietoa järjestelmien välillä täytyy olla selvillä missä muodossa tieto on. Esimerkiksi yrityksen Y-tunnus voi olla järjestelmässä A tallennettu muotoon 1234567-8 (tunnus-väliiviiva-tarkiste) ja järjestelmässä B ilman väliiviivaa ennen viimeistä numeroa.

Tietokantapohjaisessa integraatiossa tarvitaan kohdejärjestelmän tietokantakuvaus ennen SQL-lausekkeiden kirjoittamista (Nykänen, Pirkko 2015).

2.3. Avoimet integraatorajapinnat

Avoimella rajapinnalla tarkoitetaan sellaista rajapintaa jonka ominaisuudet ovat julkisia ja kaikkien käytettävissä ilman rajoittavia ehtoja. Lisäksi rajapintakuvaus ja dokumentaatio tulee olla kaikkien saatavilla ilmaiseksi. Avoin rajapinta ei kuitenkaan tarkoita, että järjestelmän kaikki tiedot olisi kenen tahansa saatavilla (Open Knowledge Finland ja COSS ry, 2014).

Rajapinnat luokitellaan datarajapinnaksi tai toiminnalliseksi rajapinnaksi. Datarajapinnan kautta pystyy vain lukemaan palvelun sisältämää dataa ja siirtämään sitä toiseen järjestelmään. Toiminnallinen rajapinta tarjoaa mahdollisuuden palvelun sisältämän datan muokkaamiseen ja uuden tiedon tuomiseen (Open Knowledge Finland ja COSS ry, 2014).

Pitkään toimineiden organisaatioiden it-järjestelmät voivat olla jopa 15-20 vuotta vanhoja. Vanhojen järjestelmien tekohelellä ei välttämättä ole ollenkaan otettu huomioon integraatiotarpeita ja järjestelmät tehtiin sillä periaatteella, että ne ovat kaikilta ulkopuolisilta suljettuja. Nykyisissä liiketoimintajärjestelmissä on kuitenkin

alettu huomaamaan rajapintojen avaamisen edut. Avoimella integraatorajapinnalla saa huomattavaa kilpailuetua sekä kiinnostavuutta toimittajana (Malinen, Teemu 2013).

Suomessa on myös viimeaikoina alettu vaatia avoimia rajapintoja julkisiin järjestelmähankintoihin muun muassa toimittajaloukkujen välttämiseksi (Kivekäs, Otso 2014).

3 ODOO-TOIMINNANOHJAUSJÄRJESTELMÄ

Odoo on avoimen lähdekoodin toiminnanohjausjärjestelmä. Aikaisemmin Odoo tunnettiin nimellä OpenERP. Nimi vaihdettiin toukokuussa 2014 juuri ennen versio 8:n julkaisua. Versio 9 julkaistiin lokakuussa 2015 (Odoo: kotisivut 2015), mutta tässä työssä käytetään vielä versiota 8.

3.1. Järjestelmän kuvaus

Järjestelmän rakenne on modulaarinen, eli se koostuu useasta keskenään integroidusta moduulista (sovelluksesta), jotka toimivat saumattomasti yhdessä ja käyttävät hyödyksi toistensa toiminnallisuuksia. Modulaarisen rakenteen ansiosta järjestelmän kokoa on helppo kasvattaa yrityksen tarpeiden mukaan lisäämällä siihen uusia moduuleita (Odoo: architecture 2015). Esimerkiksi yritys joka myy palveluja, ei välttämättä tarvitse valmistuspuolen moduuleita, jolloin ne voidaan jättää kokonaan asentamatta.

Lisättäviä ydinmoduuleita ovat esimerkiksi (Odoo: kotisivut 2015):

- taloushallinto ja kirjanpito
- verkkosivusto
- varastohallinta
- CRM
- projektinhallinta
- valmistus
- verkkokauppa
- raportointi

Lisäksi järjestelmään voidaan asentaa moduuleita, jotka muuttavat jotain näistä ydintoiminnoista tai lisäävät kokonaan uusia toimintoja (Odoo: documentation 2015). Moduuleita on saatavissa virallisesta Odoon sovelluskaupasta kirjoitushetkellä 6305

kappaletta (Odoon: kotisivut 2015). Odoolla on myös aktiivinen kehittäjäyhteisö, joka jakaa avoimen lähdekoodin Odoon-projekteja GitHub-verkkosivustolla (GitHub: Odoon Community Association 2015).

Odoon on yksi suosituimpia avoimen lähdekoodin toiminnanohjausjärjestelmiä ja sillä on ympäri maailmaa yli 600 yhteistyökumppania, jotka tuottavat Odoon-toteutuksia. Suomessa yhteistyökumppaneita on neljä (Odoon: kotisivut 2015).

3.2. Odoon arkkitehtuuri

Odoon rakenne voidaan jakaa kolmeen tasoon: tietokantatasoon, joka hoitaa tiedon varastoinnin, sovellustasoon, joka sisältää prosessoinnin ja toiminnallisuudet sekä näkymätasoon käyttöliittymän esittämiseen (Odoon: architecture 2015).

3.2.1 Tietokantataso

Odoon tietokanta pyörii PostgreSQL-tietokantapalvelimen päällä. PostgreSQL on relaatiotietokanta. Tietokanta sisältää kaiken sovelluksen tarvitseman datan sekä Odoon konfiguraatiotiedostot. Tietokantapalvelimessa voidaan käyttää ns. klusteroitua tietokantaa, mikä tarkoittaa tietokannan jakamista usealle palvelimelle (Odoon: architecture 2015).

3.2.2 Sovellustaso

Sovelluspalvelimen ORM-kerros hoitaa Odoon toimintalogiikan määrittelyn ja kommunikoinnin tietokantatason kanssa. Web-kerros puolestaan hoitaa kommunikaation sovelluspalvelimen ja verkkoselaimen välillä (Odoon: architecture 2015). Tarkemmin Odoon ORM-moottorista luvussa 3.3.

3.2.3 Näkymätaso

Näkymätaso on verkkoselaimessa toimiva Javascript-sovellus, joka kommunikoi sovelluspalvelimen kanssa JSON-RPC -protokollan välityksellä. Näkymätasolla ei ole

juuri mitään toimintalogiikkaa, vaan se on pääosin palvelimen puolella. Näkymätaso pyytää tietoa sovelluspalvelimelta ja vastauksen saatuaan näyttää tiedon. Käyttäjän toimien mukaan se myös välittää sovelluspalvelimelle pyyntöjä muokata tietoja (Odo: architecture 2015).

3.3. ORM-moottori

ORM (object-relational-mapping) tarkoittaa ohjelmistokehystä, jonka avulla olio-ohjelmoinnin tietorakenteita tallennetaan relaatiotietokantoihin (GitHub: Odo Community Association 2015). Odoossa tämä tarkoittaa Python-olioiden tallentamista PostgreSQL-tietokantaan.

Kaikki Odoon toiminnallisuus riippuu Model-olioista. Model eli malli on Pythonissa kirjoitettu luokka, joka edustaa jotain toiminnanohjauksen entiteettiä ja siihen liittyviä toimenpiteitä. Entiteetillä tässä tarkoitetaan esimerkiksi asiakas-, myyntitilaus-, valmistusmääräin tai käyttäjäolioita. Kaikki malli-ilmentymät tallennetaan PostgreSQL-tietokantaan (Liu, Ying 2014). Malliolioiden ilmentymiä käytetään sortattujen, tietokantaan tallennettujen tietueiden kautta (recordset tai yksittäisenä record).

Mallien attribuutit asetaan fields (kentät) -olioihin. Kentillä on useita eri tyyppejä, mitkä rajoittavat mitä tietoa niihin voidaan tallentaa.

Yleisimmin käytettyjä kenttätyppejä ovat esimerkiksi (Odo: rajapintakuvaus 2015):

- **Char** (käytetään tallentamaan merkkijono)
- **Text** (pidemmät tekstit, kuten esimerkiksi laskun kommentit ja tuotteen kuvaukset)
- **Integer** (kokonaisluvut)
- **Float** (desimaaliluvut)
- **Date** (päivämäärät)

- **Selection** (ennalta määritetty valikoima, kuten esimerkiksi asiakkaan maa)
- **Boolean** (totuusarvo)
- **Many2one** ("monesta yksi" -relaatio toiseen malliin, esimerkiksi laskulla on yksi asiakas)
- **One2many** ("yhdestä monta" -relaatio useampaan kappaleeseen toista mallia, esimerkiksi laskulla voi olla useita laskurivejä).

Kaikki Odoon mallioliot perivät BaseModel-luokan. BaseModel-luokka käsittää ORM-metodit tietokannan kanssa kommunikointiin.

Yleisimpiä ORM-metodeja ovat:

- **Read** (luku). Ottaa vastaan parametrinä luettavien kenttien nimet ja palauttaa listan dictionary-tietotyyppisistä avain-arvo-pareista. Yksi dictionary recordia kohden.
- **Write** (kirjoitus). Ottaa vastaan avain-arvo-pareja ja kirjoittaa arvon avainta vastaavaan kenttään recordsetissä. Kirjoittaessa Many2one-kenttään annetaan arvona relatiivisen recordin tietokantatunniste.
- **Search** (haku). Ottaa vastaan parametrejä joiden perusteella suorittaa SQL-kyselyn tietokantaan ja palauttaa löytyneen recordsetin.
- **Browse** (selaus). Ottaa vastaan selattavien recordien tietokantatunnisteet ja palauttaa vastaavan recordsetin.

Malleilla esiteltyt metodit suoritetaan recordsettien kautta ja niiden self-attribuutti on myös recordset instanssi. Malleilta voidaan myös viitata toisiin malleihin Environment-luokan avulla, johon pääsee käsiksi recordin env-attribuutin kautta.

```

from openerp import models

class Partner(models.Model):
    _inherit = "res.partner" # Perittävän mallin nimi.

    def foo(self):
        # Haku:
        # Esimerkkinä haetaan partner-malleista kaikki joiden
        # nimi on Seppo ja kaupunki Tampere.
        partners = self.search([("name", "=", "Seppo"), ("city", "=", "Tampere")])
        # löydettiin kolme partneria
        print partners
        # => res.partner(12,351,575)

        # Luku:
        # luetaan kaikilta partnereilta osoite.
        values = partners.read(["address"])
        print values
        # => [{'address': 'Hämeenkatu 17'}, {'address': 'Jokipohjantie 18'}, {'address': 'Matinkatu 1'}]

        # Kirjoitus. Vaihdetaan kaikille partnereille osoite.
        partners.write({"address": "Lapintie 4"})

        # Selaus. Esimerkkinä partner-mallin selaus
        partner_ids = [1,2,3] # lista partner id:tä
        partners = self.browse(partner_ids)
        # palauttaa recordsetin: res.partner(1,2,3)
        for partner in partners:
            print partner.name # => Maija, Ismo, Jouko

        # Environment
        # Environmenttiin päästään käsiksi mallin env-attribuutin kautta.
        # Esimerkki:
        # viittaus account.invoice-malliin
        invoice_model = self.env["account.invoice"]
        # haetaan kaikki laskut joissa partneri löytyy listasta
        invoices = invoice_model.search([("partner_id", "in", partner_ids)])
        print invoices # => account.invoice(12,46,1255,3231)

```

KUVA 1. Esimerkkejä ORM-metodien käytöstä.

4 NETVISOR-TALOUSHALLINTOPALVELU

Netvisor on Visma Solutions Oy:n tuottama taloushallinnon ohjelmistopalvelu pk-yrityksille. Netvisor on käytössä yli 13 500 suomalaisessa yrityksessä (Netvisor: kotisivut 2015). Palvelua käytetään verkkoselaimella.

4.1. Palvelut

Netvisorin sisältämät palvelut (Netvisor: palvelukuvaus 2015):

- **Myynti.** Myyntilaskujen käsittelyyn sisältyy asiakas- ja tuotehallinta, raportointi- ja seurantatyökalut sekä suoraveloitussominaisuus. Lisäpalvelumahdollisuutena tilausten käsittelyn ja varastonhallinnan ominaisuudet.
- **Osto.** Ostolaskujen käsittelyyn sisältyy toimittajien hallinta ja oletustiliöinnit. Netvisor vastaanottaa ostolaskut verkkolaskuina tai skannattuina. Ostolaskujen käsittelyn jälkeen laskut voidaan hyväksyä ja maksaa.
- **Kirjanpito.** Kirjanpidosta löytyy työkalut tiliotteiden, kirjausten, alv-laskelmien, jaksotusten ja täsmäytysten hoitamiseen. Palveluun sisältyy myös tilinpäätöksen työkalut sekä sähköinen viranomaisilmoittaminen.
- **Henkilöstöhallinto.** Henkilöstöhallinto kattaa työajankirjaukset, matkalaskutuksen ja palkanlaskennan. Palveluun kuuluu myös maksuliikenteen hoitaminen ja palkkasanomien toimittaminen sekä sähköiset viranomaisilmoitukset.
- **Raportointi.** Raportointityökaluun sisältyy muun muassa kuukausikohtaiset tulos- ja pääomarakennäköymät, kassavirtatoteumat ja -ennusteet, budjetointi sekä kasvun, kannattavuuden ja rahoituksen tunnusluvut.

4.2. Tiedonsiirto

Netvisorin palveluissa tiedonsiirrot käsittävät maksuaineiston, tilioteaineiston, suoraveloituspyynnöt ja -valtuutukset pankkeihin (Netvisor: palvelukuvaus 2015).

Tiedonsiirto yritysten ja viranomaisten välillä toimii XML-sanomina ja on toteutettu valtionvarainministeriön ohjeistuksen mukaisesti (Netvisor: palvelukuvaus 2015).

Asiakkaiden järjestelmistä tiedonsiirto toteutetaan ohjelmistorajapintakuvauksen mukaisesti. Netvisorin palveluun sisältyy työkalut siirron valvontaan, jolla voidaan valvoa siirtojen onnistumista ja poistaa niitä järjestelmästä eräkohtaisesti (Netvisor: palvelukuvaus 2015).

4.3. Verkkolaskut

Netvisor mahdollistaa myös verkkolaskujen lähetyksen ja vastaanoton Finvoice XML-muodossa (Netvisor: palvelukuvaus 2015). Finvoice verkkolasku on suomalaisten palveluntarjoajien määrittelemä ja Suomessa yleisesti käytössä oleva esitystapa (FKL: Finvoice tuotekuvaus 2015).

4.4. Tietoturva

Kaikki tietoliikenne asiakkaan selaimen ja Netvisor palveluiden välillä on SSL-salattua. Pankkiliikenne on suojattu sen mukaisesti miten pankki on aineiston suojauksen määritellyt. Liikenne on suojattu myös viranomaisille ja muille kolmansille osapuolille.

Palvelussa käyttäjälle luodaan henkilökohtaiset tunnukset ja määritetään, mihin yrityksiin käyttäjällä on oikeudet (Netvisor: palvelukuvaus 2015).

4.5. Ohjelmistorajapinta

Visma Solutions tarjoaa avoimen Web Service -ohjelmistorajapinnan, joka mahdollistaa ulkoisten järjestelmien integraatiot. Netvisorilla on myös tarjolla useita ohjelmistokumppaneiden tekemiä valmiita integraatoratkaisuja (Netvisor: kotisivut 2015). Ohjelmistorajapinta on kaksisuuntainen, eli rajapinnan kautta voidaan sekä tuoda että viedä tietoja (Netvisor: rajapintakuvaus 2015).

4.5.1 Tietojen tuominen järjestelmään

Rajapinnan kautta järjestelmään voidaan tuoda seuraavia tietoja (Netvisor: rajapintakuvaus 2015):

- myyntilaskut
- ostolaskut
- tuotteet
- asiakkaat
- kirjanpidon tositteet
- maksutoimeksiannot
- palkanlaskennan palkkaperusteet
- työajankirjaustiedot.

4.5.2 Tietojen noutaminen järjestelmästä

Rajapinta mahdollistaa seuraavien tietojen noutamisen järjestelmästä (Netvisor: rajapintakuvaus 2015):

- suoritukset
- pankkitapahtumat
- asiakkaat
- tuotteet

- laskut ja saldot
- kirjanpidon tiedot.

4.5.3 Yhteydenottotapa

Web Service -rajapinta on REST-mallin mukainen. Kommunikaatio palvelimelle tapahtuu HTTP-pyyntöillä. Järjestelmään tietoa tuodessa asiakkaan ohjelma muodostaa tuotavasta aineistosta XML-sanoman ja lähettää sen HTTP-pyyntönsä Netvisorin palvelimelle. Tietoa noudettaessa XML-sanomaa ei tarvitse lähettää pyyntönsä, vaan pyyntöä voidaan tarvittaessa rajata parametreilla (Netvisor: rajapintakuvaus 2015).

Pyynnöt kohdistetaan eri resursseihin URL:n perusteella (Netvisor: rajapintakuvaus 2015).

4.5.4 Tietoturva

Tuotantoympäristössä on käytössä HTTPS-protokolla ja testausympäristössä käytetään HTTP-protokollaa. Jokaiseen rajapintaan lähetettävään pyyntöön lisätään otsikkotietoihin tunnistautumistiedot sekä niistä laskettu MAC-koodi. Mikäli otsikkotiedot ovat puutteelliset, rajapinta palauttaa virheilmoituksen (Netvisor: rajapintakuvaus 2015).

4.5.5 Rajapintakirjasto

Netvisor tarjoaa valmiin .NET -pohjaisen rajapintakirjaston, jossa on toteutettu eri aineistojen muodostaminen ja pyyntöjen luonti sekä lähettäminen ja vastaussanomien käsittely. Kirjasto on vapaasti ladattavissa ja sitä saa vapaasti muokata sekä laajentaa (Netvisor: rajapintakuvaus 2015). Tässä työssä kirjastoa on käytetty lähinnä teknisen toteutuksen referenssinä ja esimerkkinä, mutta muuten kirjastoa ei integraatiossa käytetä.

5 INTEGRAATION SUUNNITTELU JA TOTEUTUS

5.1. Integraation suunnittelu

Integraation suunnittelussa lähdettiin siitä periaatteesta, että sille luodaan uusi Odoomoduuili, jota voidaan käyttää myös tulevaisuudessa uusissa asiakasprojekteissa. Myös moduulin jatkokehittäminen tulee ottaa huomioon ohjelmaa toteuttaessa. Koska moduulin tulee olla yleiskäyttöinen, kaikki asiakaskohtainen kustomointi jätetään toteutuksen ulkopuolelle.

5.2. Toteutus

5.2.1 Asetukset

Netvisorin rajapintaa käytettäessä tarvitaan käyttäjäkohtaisia rajapintatunnuksia. Näille tiedoille luotiin Odoossa uusi asetusvalikko Netvisoria varten.

Netvisor-asetukset sisältävät asiakkaan tunnistetiedot ja kielivalinnan. Asetuksista löytyy myös tunnistetiedot asiakkaan integraatiopartnerille sekä valinta tuotanto- ja demoympäristön välillä. Demoympäristöä käytetään uuden integraation testaamisessa ennen käyttöönottoa.

Asetussivulta löytyy myös mahdollisuus noutaa Netvisorista asiakas- tai tuoterekisteri ja päivittää niiden tiedot Odoohon.

5.2.2 Laskun vienti Odoosta Netvisoriin

Laskun lähettäminen käynnistetään Odoon-käyttöliittymässä lähettävän laskun lomakenäkymästä. Lomakkeen yläreunassa on ”Lähetä Netvisoriin” -nappi, mikäli lasku on avoin-tilassa ja sitä ei ole vielä lähetetty.

... / 0197

Print ▾ More ▾ 3 / 3

Send to Netvisor Print Register Payment Refund Invoice Cancel Invoice Draft

Invoice 0197

Customer Vizum Oy
PI 82
33101 Tampere
Finland

Invoice Date 10/02/2015

Journal Sales Journal (EUR)

Account 110200 Debtors

Fiscal Position

Invoice Lines Other Info Payments

| Product | Description | Account | Quantity | Unit Price | Taxes | Amount |
|----------|-------------|----------------------|----------|------------|---------|--------|
| X-tavara | X-tavara | 200000 Product Sales | 1.000 | 10.00 | ALV 24% | 10.00 |

Subtotal : 10.00 €

Tax : 2.40 €

Total : 12.40 €

Balance : 12.40 €

KUVA 2. Lasku Odoon käyttöliittymässä.

Netvisorin rajapintaan myyntilaskua lähetettäessä tulee ottaa huomioon rajapinnalle määritetyt pakolliset kentät.

| Netvisor | | Odoo | |
|-----------------------------|-------------------------|---------------------------|--|
| arvo | selite | Malli (Python-luokka) | kenttä |
| salesinvoicedate | Laskun päiväys | account.invoice | date_invoice |
| salesinvoiceamount | Laskun loppusumma | | amount_total |
| salesinvoicestatus | Laskun tila | | state |
| invoicingcustomeridentifier | Asiakkaan linkitystieto | res.partner | partner_id.netvisor_id partner_id.netvisor_code |
| paymenttermnetdays | Maksuehdon nettopäivät | account.payment.term.line | payment_term.line_ids.days |
| Netvisor laskurivi | | Odoo laskurivi | |
| productidentifier | Tuotteen linkitystieto | product.product | netvisor_id netvisor_code |
| productname | Tuotteen nimi | | product_id.name |
| productunitprice | Tuotteen alv prosentti | account.invoice.line | price_unit |
| productvatpercentage | Tuotteen yksikköhinta | account.tax | invoice_line_tax_id.amount |
| vatcode | Alv-koodi | | netvisor_tax_code |

| | | | |
|---------------------------------|-----------------------|----------------------|----------|
| | | | |
| salesinvoiceproductlinequantity | Tuotteen kappalemäärä | account.invoice.line | quantity |

Taulukko 1. Netvisor laskun pakolliset kentät ja niitä vastaavat kentät Odoossa

Vain rajapinnan pakolliset tiedot lähetetään, mutta ohjelmoinnissa on otettu huomioon jatkokehitys asiakaskohtaisesti. Kaikki rajapinnan kenttämääritykset on valmiiksi ohjelmoitu, joten lähetettäviä tietoja on helppo lisätä tarvittaessa.

Asiakas ja tuotteet linkitetään Netvisorissa olemassa oleviin rekistereihin. Jotta linkittäminen olisi mahdollista, täytyy tietoja lähettäessä olla selvillä Netvisoriin tallennettu tuote- ja asiakaskoodi tai tietokantatunnus. (Netvisor: rajapintakuvaus 2015)

Jos asiakas voidaan linkittää Netvisorissa oleviin asiakastietoihin, haetaan asiakkaan osoitetiedot Netvisorista. Siinä tapauksessa, että aineistossa lähetetään jotain asiakkaan osoitetietoja (esimerkiksi pelkkä postinumero), ei osoitetietoja enää haeta Netvisorin asiakaskortilta. Mikäli kohdeasiakkaalla on Netvisorissa vaihtoehtoinen laskutusasiakas, haetaan osoitetiedot automaattisesti vaihtoehtoiselta asiakkaalta, eikä laskuaineistossa lähetettyjä osoitetietoja oteta huomioon. (Netvisor: rajapintakuvaus 2015)

Laskuaineistosta tulee löytyä vähintään yksi laskurivi. Laskurivit voivat olla tuote- tai kommenttirivejä (Netvisor: rajapintakuvaus 2015). Odoossa ei kuitenkaan ole erikseen laskurivejä, jotka olisivat vain kommentteja. Tästä syystä kommenttirivien lähettäminen jätettiin tästä toteutuksesta pois. Tuote tulee linkittää Netvisorista löytyvään tuotteeseen, mutta aineistossa tuoterivin nimen, hinnan sekä muut tiedot voi vaihtaa. Tuoterivillä on myös pakollista olla Netvisorissa määritetty alv-koodi (Netvisor: rajapintakuvaus 2015). Odoossa joka laskurivillä on tiedossa Odoon verotietue, mutta siinä ei kuitenkaan ole samat alv-koodit kuin Netvisorin määrittelyssä. Odoossa verotietueelle luotiin uusi tekstikenttä, mihin käyttäjän tulee kirjoittaa kyseistä veroa vastaava alv-koodi Netvisorista.

| Lasku | | Laskentakohteet | | Varastotiedot | | | | | |
|---|--|-----------------------|------------------------|---------------------------------------|-------|-------|------|---------|------------|
| Lasku: 197, erääntynyt | | | | | | | | | |
| Laskuluetteloon Myyntireskontranäkymään | | | | Muokkaa laskua ▼ Valitse | | | | | |
| Laskun numero: | 197 | Päiväys: | 2.10.2015 | | | | | | |
| Laskutusasiakas: | Yizucor Oy | Myyjä: | Seppo Testaaja | | | | | | |
| Osoite: | PI 82 | Viitteemme: | 3503 | | | | | | |
| Osoitteen jatko: | | Viitteenne: | | | | | | | |
| Postinumero ja -toimip.: | 33101 Tampere | Maksuehto: | 14 päivää netto | | | | | | |
| Maa: | FINLAND | Eräpäivä: | 16.10.2015 | | | | | | |
| Toimitusosoite | | Viivästyskorko: | 10,5 | | | | | | |
| Toimipaikka: | Ei valittu | Huomautusalka: | 8 | | | | | | |
| Nimi: | | Toimitustapa: | Viedään | | | | | | |
| Osoite: | | Toimitusehto: | Vapaasti toimittajalla | | | | | | |
| Postinumero ja -toimip.: | | Toimitus-/lähtöpäivä: | 2.10.2016 | | | | | | |
| Maa: | | Viitenumero: | 16625026 | | | | | | |
| Valuutta: | EUR | Koronlaskentakielto: | Ei kieltoa | | | | | | |
| Verotuskäsittely: | Asiakkaan laskutusmaaryhmän mukainen käsittely | Viimeisin korkoajo: | | | | | | | |
| Kolmikantakauppa: | Ei | Y-tunnus: | 2518000-1 | | | | | | |
| | | Sopimustunnus: | | | | | | | |
| Tosite 2 | | | | | | | | | |
| Koodi | Nimike | Varasto | Määrä | Yksikköhinta | Ale-% | Alv-% | Alv | Veroton | Verollinen |
| 0009 | X-Tavara | | 1 | 10 | 0,00 | 24,00 | 2,40 | 10,00 | 12,40 |
| Asia kunnossa. | | | | | | | | | |
| | | | | | | | 2,40 | 10,00 | 10,00 |
| Verokanta 24,00 | | | | | | | 2,40 | 10,00 | |

KUVA 3. Lasku Netvisorin käyttöliittymässä.

5.2.3 XML-laskuaineiston luonti sekä tarkistus

Laskuaineisto lähetetään Netvisor rajapintaan XML-muotoisena. Aineistoa muodostettaessa tarkistetaan, että lähetettävän laskun tila on sallittu (luonnos- tai peruttuja laskuja ei lähetetä). Laskulta tarkistetaan myös, että siitä löytyy kaikki Netvisorin rajapinnassa määritellyt pakolliset kentät.

Laskun asiakkaalta ja kaikilta laskuriveiltä tarkistetaan löytyykö niille Odoosta Netvisorin linkitystieto. Jos asiakkaalla ei ole Netvisor-tunnusta, asetetaan asiakkaan linkitysarvoksi ”TEMP”. Tämä ymmärretään Netvisorin rajapinnassa väliaikaiseksi asiakkaaksi. Väliaikaiselle asiakkaalle tulee käydä vaihtamassa Netvisorin käyttöliittymässä oikea asiakas ennen kuin lasku voidaan vahvistaa (Netvisor: rajapintakuvaus 2015). Jos tuotteelta puuttuu Netvisor-tunnus, lähetään tuotteen tiedot ennen laskua Netvisoriin. Tuotetietojen onnistuneessa lähetyksestä Netvisor-rajapinta palauttaa vastauksena Netvisoriin tuotteelle luodun linkitystiedon, ja se tallennetaan Odoon tuotetietoihin. Näin toimiessa voidaan aina varmuudella linkittää laskun tuoterivit Netvisorin tuotteisiin.

Aineiston tarkistuksen jälkeen laskun tiedoista rakennetaan Netvisor-rajapinnassa kuvattu XML-aineisto.

5.2.4 Aineiston lähettäminen Netvisor-rajapintaan

Aineisto lähetetään Netvisor-rajapintaan HTTP POST-toiminnolla. Ennen lähetystä tulee pyyntöön muodostaa kappaleessa 4.5.4. esitelty otsikkotiedot tunnistautumista varten. Otsikkotietoihin lisätään asiakkaan tiedot sekä lasketaan MAC-tunniste, jonka avulla rajapinta vahvistaa kutsun tulleen otsikkotietoihin merkityltä asiakkaalta. MAC-tunniste lasketaan lisäämällä merkkijonoksi otsikkotiedoista löytyvät kentät: kutsun osoite, lähettävän ohjelman tunniste, aikaleima, asiakkaan kieli ja Netvisorin asiakaskohtainen avain (Netvisor: rajapintakuvaus 2015). Merkkijonolle lasketaan MD5-salaustekniikkaa käyttäen tarkistussumma (hash). Tämä tarkiste asetetaan MAC-tunnisteeksi. Tarkistetta ei voi purkaa alkuperäiseksi merkkijonoksi, vaan rajapinta varmistaa, että otsikoissa välitetyistä arvoista laskettu tarkiste vastaa annettua MAC-tunnistetta (Saksa, Tommi 2009). Rajapinta palauttaa virheen tunnistautumisesta, mikäli kaikkia tarvittavia otsikoita ei ole annettu.

| Arvo | Selite |
|---|--|
| X-Netvisor-Authentication-Sender | Lähettävän ohjelmiston tunniste |
| X-Netvisor-Authentication-PartnerId | Integraatiopartnerin tunniste |
| X-Netvisor-Authentication-Timestamp | Aikaleima |
| X-Netvisor-Interface-Language | Asiakkaan kieli Netvisor-käyttöliittymässä |
| X-Netvisor-Authentication-TransactionId | Aineiston yksilöivä tunniste |
| X-Netvisor-Authentication-CustomerId | Asiakkaan tunniste |
| X-Netvisor-Organization-ID | Asiakkaan yrityksen Y-tunnus |
| X-Netvisor-Authentication-MAC | MAC-tunniste |

Taulukko 2. Kutsun otsikot rajapinnan tunnistautumista varten.

Rajapinta vastaa jokaiseen integraatiokutsuun vakioidulla sanomavastauksella, josta ilmenee tapahtuiko kutsun käsittelyssä virheitä. Virheellisissä kutsuissa on myös selite, miksi kutsu hylättiin. Tuodessa dataa rajapintaan, palauttaa rajapinta myös tuodun aineiston saaman Netvisor-tunnuksen (Netvisor: rajapintakuvaus 2015).

Netvisor-tunnus tallennetaan Odoossa laskun tietoihin, jotta myöhemmin on helpompi löytää lähetetty lasku Netvisorista. Onnistuneen lähetyksen jälkeen Odoon laskulle merkitään myös lähetyksen päivämäärä ja merkataan lasku lähetetyksi, jotta sitä ei voi

lähettää uudelleen. Jos lähetyksessä tapahtui virhe, näytetään rajapinnan virhekoodi ja selite käyttäjälle.

6 POHDINTA

Opinnäytetyöraportin kirjoittaminen oli varsin hidas ja pitkä prosessi, joka välillä unohtuikin useaksi kuukaudeksi. Työn ohjelmalliseen tuotokseen olen silti tyytyväinen. Se täyttää alussa määritellyt tavoitteet ja on ohjelmoinnin kannalta toiminut itselleni hyvänä referenssinä muissa Odoon integraatitoteutuksissa. Netvisor-integraatio oli itselleni ensimmäinen laskutusjärjestelmään tehty integraatio. Myöhemmin tekemissäni integraatioissa on ollut helpompi aloittaa itse ohjelmointityö, kun tästä työstä saadun kokemuksen ansiosta integraation suunnittelu on paljon selkeämpi prosessi. Myös varsinaisen python-moduulin rakenne on ollut myöhemmissä toteutuksissa selkeämpi, kun tietää heti aloittaessa mitä on tekemässä. Muita integraatioita tehdessä olen myös huomannut kuinka hyvä Netvisorin rajapinnan dokumentaatio on verrattuna muihin alan toimijoihin.

LÄHTEET

Odoo. 2015. Odoo kotisivut. Luettu 13.10.2015. <https://www.odoo.com/>

Netvisor. 2015. Palvelukuvaus. Ladattu 13.10.2015.

https://netvisor.zendesk.com/hc/fi/article_attachments/201132378/Visma_Solution_Oy_Netvisor_palvelukuvaukset_2015.pdf

Odoo. 2015. Versio 8.0 API. Luettu 13.10.2015.

<https://www.odoo.com/documentation/8.0/reference/orm.html>

Odoo. 2015. Architecture. Luettu 13.10.2015. [http://odoo-](http://odoo-docs.readthedocs.org/en/latest/02_architecture.html)

[docs.readthedocs.org/en/latest/02_architecture.html](http://odoo-docs.readthedocs.org/en/latest/02_architecture.html)

Netvisor. 2015. Ohjelmistorajapinta. Luettu 17.10.2015.

<https://netvisor.zendesk.com/hc/fi/articles/201868436-Ohjelmistorajapinta>

Nykänen, Pirkko. 2015. Tietojärjestelmien integraatiosta ja integraation suunnittelusta
Ladattu 13.10.2015.

http://www.uta.fi/sis/tie/tjsuom/index/TJSUM_Luento5_2015_PirkkoNyk%C3%A4nen.pdf

Teemu Malinen. 2013. Blogi kirjoitus. Luettu 15.10.2015.

<https://www.sofokus.com/blogi/mita-tarkoittaa-ohjelmiston-integraatio-ja-miksi-se-voituplata-kilpailuetusi/>

Netvisor. 2015. Netvisor kotisivut. Luettu 17.10.2015. <http://www.netvisor.fi/>

Finanssialan Keskusliitto. 2015. Finvoice tuotekuvaus. Luettu 17.10.2015.

<https://www.fkl.fi/teemasivut/finvoice/finvoice-tuotekuvaus/Sivut/default.aspx>

Odoo. 2015. Odoo documentation. Luettu 17.10.2015.

<https://www.odoo.com/documentation/8.0/howtos/backend.html>

GitHub. 2015. Odoo Community Association. Luettu 18.10.2015.

<https://github.com/OCA>

Ying Liu. 2014. Blogi-kirjoitus. Luettu 20.10.2015.

<http://www.mindissoftware.com/2014/11/07/Understand-Odoo-Model-Part1/>

Tommi Saksa, 2009. HAMK-opetusmateriaali. Luettu 15.10.2016

<https://wiki.hamk.fi/display/materiaalit/md5-funktio+merkkijonojen+salaukseen>

Open Knowledge Finland ja COSS ry, Verkkosivu, Luettu 2.11.2016

<http://avoinrajapinta.fi/2014/10/11/avoin-rajapinta/>